Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Title of Thesis

Master Thesis

S. Tudent

January 19, 2038

Advisors: Prof. Dr. A. D. Visor, Dr. P. Ostdoc

Department of Computer Science, ETH Zürich

Abstract

Abstract goes here.

Contents

Chapter 1

Introduction

Blabla

1.1 Features

The template is divided into TeX files as follows:

1. `thesis.tex` is the main file.
2. Here we introduce the ToxCast pipeline (tcpl), a novel R extension (R Core Team, 2016), to provide storage, normalization, dose–response modeling and visualization solutions for HTS screening efforts.tcpl provides functionality to identify potentially active compounds (positive hit-calls) for both single- and multiple-concentration testing paradigms. Multiple-concentration processing also includes dose-response modeling to give potency and efficacy estimates, and categorization of each concentration series to better identify potential false positive and false negative results.
3. Often controversial uncertainty factors must be applied to account for differences between test animals and humans. Finally, use of animals in testing is expensive and time consuming, and it sometimes raises ethical issues.
4. Tropsha10demonstrated that the presence orabsence of structural errors in a library and the choice of descriptors had agreater impact on performance than model optimization. Hence, there is aneed to pay attention to systematic chemical curation protocols prior tomodeling. Fourche et al.14provide a reproducible workflow for cleaning upchemical data prior to developing a model.
5. Regardless of how generalized a model may appear to be following valid-ation, it is impractical to consider the model applicable to the

entire chem-ical space. The predictions made by models on new compounds withdescriptor values outside the training data descriptor (feature) space maynot be reliable. It is therefore necessary to know the boundary withinwhich the model can extrapolate reliably. The applicability domain (AD) defines the scope and limitations of a model. AD attempts to define thedegree of generalization of the model by highlighting the range of chemicalstructures for which the model is considered to be reliably applicable.94,95Predictions of compounds outside a models AD cannot be consid-ered reliable.

6. Distance-based methods use distance measures (e.g. Tanimoto or Euclidean) to calculate the distance between a new compound and its k-nearest neighbors (or the centroid of the training set). A threshold, basedon distance, is used to determine if the new compound is within the ADor not. Predictions of any compound beyond the threshold are consideredto be unreliable. The downside of this method is that the threshold value isoften arbitrary. Using the Enalos module, KNIME provides a graphicaluser interface to generate the AD domain based on Euclidean Distance andLeverage. One other type of non-descriptor method is the structural fragment-based method, which requires that all structural fragments in the newmolecule be present in the training set.

7. Ensemble learners coupledwith resampling include UnderBagging, 77SMOTEBagging, 78SMOTEBoost,79 and EUSBoost80 (which is considered an improvement over RUSBoost).

8. Metrics such as Accuracyand even AUROC tend to be rather optimistic82when dealing with180G. IDAKWO ET AL.imbalanced data. AUPRC and other metrics such as Balanced Accuracy,Sensitivity, and Specificity appear to provide a better or at least comple-mentary evaluation for imbalanced classifiers

9. With the large number of available descriptors,26datasets often suffer fromthe"curse of dimensionality (problems caused by performing predictionsin a very large feature space) and the so-called"large p, small n

10. Models with fewer descriptors are easier to interpret, lesscomputationally expensive, higher performing for new molecules, and lessprone to overfitting/overtraining

11. In other words, models trained on a very small set of molecules that aredescribed with a very large set of descriptors tend to be prone to overfit-ting.37An overfitted model can mistake small fluctuations for importantvariance in the data, which can result in significant prediction

errors.Identifying reliable descriptors for establishing this relationship can pose aserious challenge.

12. The process does not alter the original repre-sentation of the descriptors, thus maintaining the physical meanings andallowing for interpretability

13. Feature selection involves picking a subset of features by eliminatin-girrelevant and redundant descriptors, yielding the best possible per-formancebased on a selection criterion. The process does not alter the original repre-sentation of the descriptors, thus maintaining the physical meanings andallowing for interpretability. Feature selection techniques can be classifiedinto filter, wrapper, and embedded meth-ods.26,40,41Filters work withouttaking the classifier into consideration. They rely on measures of the gen-eral characteristics of the training data,

14. Multiple fingerprint predictions were obtained for one chemical if both negative and positive mode spectra were available or by different research groups or with different LC-HRMS parameters. For valida-tion, fingerprints were calculated using SIRIUS+CSI:FingerID version 4.9.5. (41,42) All of the chosen parameters for SIRIUS+CSI:FingerID are described in the SI section "Fingerprints Calculated with SIRIUS Software". https://pubs.acs.org/doi/full/10.1021/acs.est.2c02536

15. The US EPA ToxCast program generates high-throughput screening (HTS) bioactivity data for use in various predictive toxicology appli-cations 32, 35. The ToxCast data pipeline (tcpl) [17] has been used to normalize and curve-fit data for nearly 1400 assay end points, thus enabling first-tier data processing of heterogeneous bioactivity data from HTS. For each chemical sample:assay end point pair with at least 4 concentrations available, the tcpl curve-fitting procedure attempts to fit a Hill, a gain-loss, and a constant model, with the model selec-tion based on a maximum likelihood estimate [17]. The tcpl analysis generates concentration–response parameters on the basis of the win-ning model, including the 50 percent activity concentration (AC50), the potency estimate. There are multiple sources of potential variability in these AC50s, resulting from biological variance, experimental error, or curve-fitting procedures. One method for quantifying uncertainty in ToxCast curve-fitting is called toxboot, an R package extension of tcpl that implements smooth nonparametric bootstrapping (a statistical method that uses resampling and added noise to determine uncertainty in a series) 76, 77. This addition of random, normally distributed noise to the series allows one to be more confident in the winning model if similar models are produced in each iteration. By resampling and adding normally distributed noise over many iterations, a general pic-ture of the confidence in a curve fit can be ascertained. Two examples

of the quantitative measures of uncertainty produced by toxboot are hit percent and an AC50 confidence interval. Hit percent is the number of active hit-calls of the total number of resamples, currently with 1000 resamples per curve. This is potentially informative because the binary hit-call for a borderline response, positive or negative, is particularly susceptible to minor fluctuations in the data, especially for weak or borderline responses. With a more continuous statistic (ranging from 0 to 1), such as hit-call percent, the borderline hit-calls may be more obvious; currently, roughly 61 percent of positive hit-calls (hitcall equals 1) correspond to a hit percent of 100 in the latest ToxCast database release [16]. Additionally, tcpl can be executed upon each resampled set and generate an AC50 (or AC10, AC20, area under the curve, etc.) and further generate a median value and a 95 percent confidence interval, on the basis of the results. These approaches could be used as a method to determine the confidence in potency estimates derived from concentration series. A summary of curve-fitting uncertainty information is available in level 7 of the latest ToxCast database release, invitrodb version 3.1 [16].

16. The issue of quantitative uncertainty in fitting in vitro activity data is not unique to ToxCast data, and different approaches have been developed to handle this uncertainty. Previously, the US EPA's Benchmark Dose software, used commonly in modeling dose–response information from in vivo toxicity studies to define PODs, has been adapted for use in modeling gene expression data sets as BMDExpress 3, 82. BMDExpress enables automated analysis of continuous transcriptomic data via identification of genes that demonstrate significant dose–response behavior, followed by curve-fitting with statistical models that can be used to define a confidence interval around a potency estimate for significant changes in expression for a given gene. The upper and lower bounds on the confidence interval for a BMD thus give a sense of the uncertainty in fitting the available transcriptomic data. A commonality between the BMDExpress and toxboot approaches, both suited to HTS data of different types, is the desire to communicate a confidence interval around the potency estimate. BMDExpress is flexible in that the upper and lower bounds (BMDU and BMDL, respectively) of the confidence interval around a calculated potency value can be modified, allowing for differing levels of uncertainty in the analysis. Using default parameters, for a differentially expressed gene, BMDExpress will calculate a 95 percent confidence interval around a BMD for a 10 percent increase or decrease in response compared to the background of the control samples.

17. hi

18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.

48.

49.

50.

Chapter 2

Writing scientific texts in English

This chapter was originally a separate document written by Reto Spöhel. It is reprinted here so that the template can serve as a quick guide to thesis writing, and to provide some more example material to give you a feeling for good typesetting.

2.1 Basic writing rules

The following rules need little further explanation; they are best understood by looking at the example in the booklet by Knuth et al., §2–§3.

Rule 2.1 Write texts, not chains of formulas.

More specifically, write full sentences that are logically interconnected by phrases like 'Therefore', 'However', 'On the other hand', etc. where appropriate.

Rule 2.2 Displayed formulas should be embedded in your text and punctuated with it.

In other words, your writing should not be divided into 'text parts' and 'formula parts'; instead the formulas should be tied together by your prose such that there is a natural flow to your writing.

2.2 Being nice to the reader

Try to write your text in such a way that a reader enjoys reading it. That's of course a lofty goal, but nevertheless one you should aspire to!

Rule 2.3 Be nice to the reader.

Give some intuition or easy example for definitions and theorems which might be hard to digest. Remind the reader of notations you introduced many pages ago – chances are he has forgotten them. Illustrate your writing with diagrams and pictures where this helps the reader. Etc.

Rule 2.4 Organize your writing.

Think carefully about how you subdivide your thesis into chapters, sections, and possibly subsections. Give overviews at the beginning of your thesis and of each chapter, so the reader knows what to expect. In proofs, outline the main ideas before going into technical details. Give the reader the opportunity to 'catch up with you' by summing up your findings periodically.

Useful phrases: 'So far we have shown that . . . ', 'It remains to show that . . . ', 'Recall that we want to prove inequality (7), as this will allow us to deduce that . . . ', 'Thus we can conclude that . . . . Next, we would like to find out whether . . . ', etc.

Rule 2.5 Don't say the same thing twice without telling the reader that you are saying it twice.

Repetition of key ideas is important and helpful. However, if you present the same idea, definition or observation twice (in the same or different words) without telling the reader, he will be looking for something new where there is nothing new.

Useful phrases: 'Recall that [we have seen in Chapter 5 that] . . . ', 'As argued before / in the proof of Lemma 3, . . . ', 'As mentioned in the introduction, . . . ', 'In other words, . . . ', etc.

Rule 2.6 Don't make statements that you will justify later without telling the reader that you will justify them later.

This rule also applies when the justification is coming right in the next sentence! The reasoning should be clear: if you violate it, the reader will lose valuable time trying to figure out on his own what you were going to explain to him anyway.

Useful phrases: 'Next we argue that . . . ', 'As we shall see, . . . ', 'We will see in the next section that . . . , etc.

2.3 A few important grammar rules

Rule 2.7 There is (almost) *never* a comma before 'that'.

It's really that simple. Examples:

> We assume that . . .
> *Wir nehmen an, dass . . .*

It follows that ...
Daraus folgt, dass ...

'thrice' is a word that is seldom used.
'thrice' ist ein Wort, das selten verwendet wird.

Exceptions to this rule are rare and usually pretty obvious. For example, you may end up with a comma before 'that' because 'i.e.' is spelled out as 'that is':

For $p(n) = \log n/n$ we have ... However, if we choose p a little bit higher, that is $p(n) = (1+\varepsilon)\log n/n$ for some $\varepsilon > 0$, we obtain that...

Or you may get a comma before 'that' because there is some additional information inserted in the middle of your sentence:

Thus we found a number, namely n_0, that satisfies equation (13).

If the additional information is left out, the sentence has no comma:

Thus we found a number that satisfies equation (13).

(For 'that' as a relative pronoun, see also Rules 2.9 and 2.10 below.)

Rule 2.8 There is usually no comma before 'if'.

Example:

A graph is not 3-colorable if it contains a 4-clique.
Ein Graph ist nicht 3-färbbar, wenn er eine 4-Clique enthält.

However, if the 'if' clause comes first, it is usually separated from the main clause by a comma:

If a graph contains a 4-clique, it is not 3-colorable .
Wenn ein Graph eine 4-Clique enthält, ist er nicht 3-färbbar.

There are more exceptions to these rules than to Rule 2.7, which is why we are not discussing them here. Just keep in mind: don't put a comma before 'if' without good reason.

Rule 2.9 Non-defining relative clauses have commas.

Rule 2.10 Defining relative clauses have no commas.

In English, it is very important to distinguish between two types of relative clauses: defining and non-defining ones. This is a distinction you absolutely need to understand to write scientific texts, because mistakes in this area actually distort the meaning of your text!

It's probably easier to explain first what a *non-defining* relative clause is. A non-defining relative clauses simply gives additional information *that could also be left out* (or given in a separate sentence). For example, the sentence

> The WeirdSort algorithm, which was found by the famous mathematician John Doe, is theoretically best possible but difficult to implement in practice.

would be fully understandable if the relative clause were left out completely. It could also be rephrased as two separate sentences:

> The WeirdSort algorithm is theoretically best possible but difficult to implement in practice. [By the way,] WeirdSort was found by the famous mathematician John Doe.

This is what a non-defining relative clause is. *Non-defining relative clauses are always written with commas.* As a corollary we obtain that you cannot use 'that' in non-defining relative clauses (see Rule 2.7!). It would be wrong to write

> ~~The WeirdSort algorithm, that was found by the famous mathematician John Doe, is theoretically best possible but difficult to implement in practice.~~

A special case that warrants its own example is when 'which' is referring to the entire preceding sentence:

> Thus inequality (7) is true, which implies that the Riemann hypothesis holds.

As before, this is a non-defining relative sentence (it could be left out) and therefore needs a comma.

So let's discuss *defining* relative clauses next. A defining relative clause tells the reader *which specific item the main clause is talking about*. Leaving it out either changes the meaning of the sentence or renders it incomprehensible altogether. Consider the following example:

> The WeirdSort algorithm is difficult to implement in practice. In contrast, the algorithm that we suggest is very simple.

Here the relative clause 'that we suggest' cannot be left out – the remaining sentence would make no sense since the reader would not know which algorithm it is talking about. This is what a defining relative clause is. *Defining relative clauses are never written with commas.* Usually, you can use both 'that' and 'which' in defining relative clauses, although in many cases 'that' sounds better.

As a final example, consider the following sentence:

> For the elements in $\mathcal{B}$ which satisfy property (A), we know that equation (37) holds.

This sentence does not make a statement about all elements in $\mathcal{B}$, only about those satisfying property (A). The relative clause is *defining*. (Thus we could also use 'that' in place of 'which'.)

Table 2.1: Things you (usually) don't say

~~It holds (that) ...~~	We have ...	*Es gilt ...*
('Equation (5) holds.' is fine, though.)		
~~x fulfills property $\mathcal{P}$.~~	x satisfies property $\mathcal{P}$.	*x erfüllt Eigenschaft $\mathcal{P}$.*
~~in average~~	on average	*im Durchschnitt*
~~estimation~~	estimate	*Abschätzung*
~~composed number~~	composite number	*zusammengesetzte Zahl*
~~with the help of~~	using	*mit Hilfe von*
~~surely~~	clearly	*sicher, bestimmt*
~~monotonously increasing~~	monotonically incr.	*monoton steigend*
(Actually, in most cases 'increasing' is just fine.)		

In contrast, if we add a comma the sentence reads

> For the elements in $\mathcal{B}$, which satisfy property (A), we know that equation (37) holds.

Now the relative clause is *non-defining* – it just mentions in passing that all elements in $\mathcal{B}$ satisfy property (A). The main clause states that equation (37) holds for *all* elements in $\mathcal{B}$. See the difference?

2.4 Things you (usually) don't say in English – and what to say instead

Table 2.1 lists some common mistakes and alternatives. The entries should not be taken as gospel – they don't necessarily mean that a given word or formulation is wrong under all circumstances (obviously, this depends a lot on the context). However, in nine out of ten instances the suggested alternative is the better word to use.

Chapter 3

Typography

3.1 Punctuation

Rule 3.1 Use opening (‘) and closing (’) quotation marks correctly.

In LaTeX, the closing quotation mark is typed like a normal apostrophe, while the opening quotation mark is typed using the French *accent grave* on your keyboard (the *accent grave* is the one going down, as in *frère*).

Note that any punctuation that *semantically* follows quoted speech goes inside the quotes in American English, but outside in Britain. Also, Americans use double quotes first. Oppose

> “Using ‘lasers,’ we punch a hole in … the Ozone Layer,” Dr. Evil said.

to

> ‘Using “lasers”, we punch a hole in … the Ozone Layer’, Dr. Evil said.

Rule 3.2 Use hyphens (-), en-dashes (–) and em-dashes (—) correctly.

A hyphen is only used in words like ‘well-known’, ‘3-colorable’ etc., or to separate words that continue in the next line (which is known as hyphenation). It is entered as a single ASCII hyphen character (`-`).

To denote ranges of numbers, chapters, etc., use an en-dash (entered as two ASCII hyphens `--`) with no spaces on either side. For example, using Equations (1)–(3), we see…

As the equivalent of the German *Gedankenstrich*, use an en-dash with spaces on both sides – in the title of Section 2.4, it would be wrong to use a hyphen instead of the dash. (Some English authors use the even longer emdash (—)

instead, which is typed as three subsequent hyphens in LaTeX. This emdash is used without spaces around it—like so.)

3.2 Spacing

Rule 3.3 Do not add spacing manually.

You should never use the commands `\\` (except within tabulars and arrays), `\␣` (except to prevent a sentence-ending space after Dr. and such), `\vspace`, `\hspace`, etc. The choices programmed into LaTeX and this style should cover almost all cases. Doing it manually quickly leads to inconsistent spacing, which looks terrible. Note that this list of commands is by no means conclusive.

Rule 3.4 Judiciously insert spacing in maths where it helps.

This directly contradicts Rule 3.3, but in some cases TeX fails to correctly decide how much spacing is required. For example, consider

$$f(a,b) = f(a+b,a-b).$$

In such cases, inserting a thin math space `\,` greatly increases readability:

$$f(a,b) = f(a+b,\, a-b).$$

Along similar lines, there are variations of some symbols with different spacing. For example, Lagrange's Theorem states that $|G| = [G : H]|H|$, but the proof uses a bijection $f\colon aH \to bH$. (Note how the first colon is symmetrically spaced, but the second is not.)

Rule 3.5 Learn when to use `\␣` and `\@`.

Unless you use 'french spacing', the space at the end of a sentence is slightly larger than the normal interword space.

The rule used by TeX is that any space following a period, exclamation mark or question mark is sentence-ending, except for periods preceded by an upper-case letter. Inserting \ before a space turns it into an interword space, and inserting `\@` before a period makes it sentence-ending. This means you should write

```
Prof.\ Dr.\ A. Steger is a member of CADMO\@.
If you want to write a thesis with her, you
should use this template.
```

which turns into

> Prof. Dr. A. Steger is a member of CADMO. If you want to write a thesis with her, you should use this template.

The effect becomes more dramatic in lines that are stretched slightly during justification:

> Prof. Dr. A. Steger is a member of CADMO. If you

Rule 3.6 Place a non-breaking space (~) right before references.

This is actually a slight simplification of the real rule, which should invoke common sense. Place non-breaking spaces where a line break would look 'funny' because it occurs right in the middle of a construction, especially between a reference type (Chapter) and its number.

3.3 Choice of 'fonts'

Professional typography distinguishes many font attributes, such as family, size, shape, and weight. The choice for sectional divisions and layout elements has been made, but you will still occasionally want to switch to something else to get the reader's attention. The most important rule is very simple.

Rule 3.7 When emphasising a short bit of text, use `\emph`.

In particular, *never* use bold text (`\textbf`). Italics (or Roman type if used within italics) avoids distracting the eye with the huge blobs of ink in the middle of the text that bold text so quickly introduces.

Occasionally you will need more notation, for example, a consistent typeface used to identify algorithms.

Rule 3.8 Vary one attribute at a time.

For example, for WeirdSort we only changed the shape to small caps. Changing two attributes, say, to bold small caps would be excessive (LaTeX does not even have this particular variation). The same holds for mathematical notation: the reader can easily distinguish g_n, $G(x)$, $\mathcal{G}$ and G.

Rule 3.9 Never underline or uppercase.

No exceptions to this one, unless you are writing your thesis on a typewriter. Manually. Uphill both ways. In a blizzard.

3.4 Displayed equations

Rule 3.10 Insert paragraph breaks *after* displays only where they belong. Never insert paragraph breaks *before* displays.

LaTeX translates sequences of more than one linebreak (i.e., what looks like an empty line in the source code) into a paragraph break in almost all contexts. This also happens before and after displays, where extra spacing is inserted to give a visual indication of the structure. Adding a blank line in these places may look nice in the sources, but compare the resulting display

$$a = b$$

to the following:

$$a = b$$

The first display is surrounded by blank lines, but the second is not. It is bad style to start a paragraph with a display (you should always tell the reader what the display means first), so the rule follows.

Rule 3.11 Never use `eqnarray`.

It is at the root of most ill-spaced multiline displays. The *amsmath* package provides better alternatives, such as the `align` family

$$\begin{aligned} f(x) &= \sin x, \\ g(x) &= \cos x, \end{aligned}$$

and `multline` which copes with excessively long equations:

$$\begin{multline} \mathrm{P}\big[X_{t_0} \in (z_0, z_0 + dz_0], \dots, X_{t_n} \in (z_n, z_n + dz_n]\big] \\ = \nu(dz_0) K_{t_1}(z_0, dz_1) K_{t_2 - t_1}(z_1, dz_2) \cdots K_{t_n - t_{n-1}}(z_{n-1}, dz_n). \end{multline}$$

3.5 Floats

By default this style provides floating environments for tables and figures. The general structure should be as follows:

```
\begin{figure}
  \centering
  % content goes here
  \caption{A short caption}
  \label{some-short-label}
\end{figure}
```

Note that the label must follow the caption, otherwise the label will refer to the surrounding section instead. Also note that figures should be captioned at the bottom, and tables at the top.

The whole point of floats is that they, well, *float* to a place where they fit without interrupting the text body. This is a frequent source of confusion and changes; please leave it as is.

Rule 3.12 Do not restrict float movement to only 'here' (`h`).

If you are still tempted, you should avoid the float altogether and just show the figure or table inline, similar to a displayed equation.

Chapter 4

Example Chapter

Dummy text.

4.1 Example Section

Dummy text.

4.1.1 Example Subsection

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Appendix A

Dummy Appendix

You can defer lengthy calculations that would otherwise only interrupt the flow of your thesis to an appendix.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.